

課題7：テキストファイルの読み込み

HTML5 ではローカルファイルの内容を読み込むことができます。

Javascript のおもしろいところは ajax のように裏に隠れて実行を続けることができるところで、ファイルの読み込みも非同期で行います。readAsText メソッドが読み込みを行いますが、大きなファイルであっても最後まで読み込みをして他の処理の実行を止めることはありません。じょじょに読み込み、onprogress で total と loaded からどこまで読み込んでいるかの途中経過を表示できます。これは先に紹介した、progress バーを使用します。

また今回は “text/plain”,”text/html”,”text/xml”,”text/css”,”text/rtf”のテキストファイルのみ読むように制限を加えています。type の頭5文字を見て text 系でなければ return しています。

【index.html】

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>選択したテキストファイルの内容を表示</title>
    <script src="js/file.js"></script>
  </head>
  <body>
    <h1>選択したテキストファイルの内容を表示</h1>
    <form>
      <input type="file" id="filedata" accept="text/*">
      <input type="button" id="read" value="選択したファイル内容を表示">
      <input type="button" id="stopButton" value="読み込みを中断">
    </form>
    <progress id="loadstatus" max="100" value="0">Loading...</progress>
    <div id="fileinfo"></div>
    <div id="contents"></div>
  </body>
</html>
```

【file.js】

```
// 選択したテキストファイルの内容を表示
window.addEventListener("load", function(){
  // File API が使えるか調べる
```

```

if (!window.File){
    ele.innerHTML = "File API が使用できません";
    return;
}
// 情報を表示する領域の要素
var ele = document.getElementById("fileinfo");
// 進捗状況を表示するプログレスバーの要素を特定
var prog = document.getElementById("loadstatus");
// ファイルを読み込むための File Reader オブジェクトを入れる変数
var reader;

// 「保存する」ボタンがクリックされた時の処理
document.getElementById("read").addEventListener("click", function(){
    var textFile = document.getElementById("filedata").files[0];
    // 選択されたファイル情報
    ele.innerHTML = "<hr>ファイル名：" + textFile.name;
    ele.innerHTML += "<br>ファイルサイズ：" + textFile.size + " バイト";
    ele.innerHTML += "<br>MIME Type：" + textFile.type;
    ele.innerHTML += "<hr>";
    // テキストかどうか調べる
    if (textFile.type.indexOf("text/") != 0){
        ele.innerHTML += "選択したファイルはテキスト形式ではありません";
        return;
    }
    // テキストファイルなら処理を行う
    reader = new FileReader();
    reader.onload = function(evt){
        var totalData = evt.total;
        prog.innerHTML = "100% (" + totalData + "/" + totalData + " バイト)";
        prog.value = 100;
        ele.innerHTML += "読み込み完了";
        var text = evt.target.result;
        var text = text.substr(0, 100); // 先頭 100 文字だけ表示
        document.getElementById("contents").textContent = text;
        // document.getElementById("contents").innerText = text;
    }
    reader.onerror = function(evt){
        var errorNo = evt.target.error.code

```

```

        ele.innerHTML += "エラー発生："+errorNo;
    }
    reader.onabort = function(evt){
        ele.innerHTML += "読み込みが中断されました";
    }
    reader.onprogress = function(evt){
        var loadData = evt.loaded;
        var totalData = evt.total;
        var per = (loadData/totalData) * 100;
        per = per.toFixed(1);    // 小数点第一位までの表示にする
        prog.innerHTML = per+"% ("+loadData+"/"+totalData+" バイト)";
        prog.value = per;
    }
    reader.readAsText(textFile, "utf-8");
}, true);
// 読み込み停止ボタン
document.getElementById("stopButton").addEventListener("click", function(){
    reader.abort();
}, true);
}, true);

```

かなり大きなテキストファイルを用意しないと途中経過はみられない。